UNIVERSITY OF TECHNOLOGY, JAMAICA

SCHOOL OF COMPUTING AND INFORMATION TECHNOLOGY

COMPUTER SCIENCE MAJOR
THEORY OF COMPUTATION (CIT3006) - GROUP PROJECT
DRAFT PROJECT DOCUMENT

Documentation Due Date: ***March 7, 2023***
Presentation: ***April 4, 2023***

This project aims to simulate the computation of the Turing and register machines.

A Turing machine (TM) consists of a tape of infinite length and a read-write head which can move either left or right across the tape. For each transition, the machine checks the current state and the character on the tape below the head, then changes to a new state, writes a new character on the tape, and moves the head one space left or right. The machine stops after the transferring to accept or reject state.

A register machine (RM) is an abstract machine that simulates a TM's behaviour. In contrast to the tape and head of the TM, the register machine has an infinite number of registers, and the content of each register holds a single positive integer which ranges from zero to infinity. The program of a register machine consists of a finite list of instructions that is a combination of three functions increment, decrement, and check if zero.

Project Instructions

Develop a program utilizing any programming language of your choice that illustrates the acceptance or rejection of an input string to a vending machine.

Basic Idea

The vending machine has four items (napkins, knives, forks & spoons) and accepts only three types of Jamaican coins each noted by a symbol: $5 (**α**), $10 (**β**) and $20 (**γ**)

(You may change symbols for money if needed)

This vending machine can hold 20 of each item at the start, but each has a different price:

Napkin (**N**) = $10	Knife (**K**) = $25	Fork (**F**) = $15	Spoon (**S**) = $20

The customer will enter money acceptable to the vending machine, and then specify which item(s) they want. The machine will then verify input and either:

1. Dispense requested item(s)
2. Refund money based on insufficient funds.
3. Refund money if the stock for an item is lower than the request.
4. Reject string based on incorrect user input.

For instance, the inputs below will yield the following results:

- **ααβγFNF** – outputs 2 forks and a napkin
- **γβαSK** – outputs spoon but doesn't dispense knife due to inefficient funds (refund $15)
- **δNKF** – rejects input based on an unrecognizable character in the string.
- **γN** – this is an acceptable string but if stock is low item won't be dispensed (refund $20)

The owner of the vending machine should be able to collect funds and restock the machine based on the following input **NKSFFSKS**. Once this input is received it will calculate the funds obtained from sales and specify what is left in stock. This information should then be recorded in a file with a timestamp. The till will then return to zero along with the starting inventory of each item.

As the developer of the simulator program, you should specify how to halt the machine.

Marking Scheme:

Documentation: The following pieces are required for submission within a single file on Moodle.

1. Outline the operating parameter that limits your application. For instance:
   - the type of Turing machine used and reason **[3 marks]**
   - alphabet utilized (if different from what was prescribed)
   - what data types are used and how do they relate to the components of the automaton being implemented **[3 marks]**
   - any other circumstances that were not considered in design **[2 marks]**

   Your operating parameter should not contradict requirements.

2. Automaton Design: The formal specification of the Turing Machine **[6 marks]**

3. Algorithm for the Turing machines (note algorithm is not the code). **[8 marks]**

4. For each register machine used provide a program/graphical representation with their initial and final configuration. **[16 marks]**

5. For each algorithm determine the time and space class. **[5 marks]**

6. Write a guide with associated images that provides instructions for using a program, also providing a clear description of who did what. **[3 marks]**

Presentation: Using a **<u>single</u>** application, simulate the vending machine operations described previously and be prepared to answer any questions on your program. Presentation time and date will be scheduled for each group within the 13th week using the following grading scheme.

- User input: **[6 marks]**
  - accept string if it belongs to language and specifies action taken.
  - reject string if it does not belong to the language and provide a reason for doing so.

- You have free range in displaying your creativity within the program interface, but it should represent the manoeuvres of the Turing and register machines. **[20 marks]**

- Presentation and readability of code. **[4 marks]**

Submission & Penalties:

1. The project should be done in groups with NO MORE than four (4) students per group. Failure to comply will result in the assessment not being accepted and group members receiving 0(NS). **<u>No individual projects will be accepted.</u>**
2. Completed implementation must run and source code submitted with the project report. Projects which do not run will not receive a passing grade.
3. The names of ALL participating group members and ID numbers should be within the documentation submitted on Moodle. Students will be graded for their contribution; therefore, **ALL reports MUST specify who completed specified individual components.**
4. A 5–7-minute presentation should be done during your tutorial session (DATE) to complete the grading of the project.
5. 3 marks per day will be deducted for late submission. **3 days late – project will not be accepted.**
6. Plagiarism is considered a very serious offence by the university and will be penalized as outlined in the Student Handbook (see Regulation 5).